

# Server API Document

Yi Zhao

2014/10/10

## Contents

<b>1</b>	<b>Main functions</b>	<b>2</b>
1.1	Submit a Crisper-X request . . . . .	2
1.2	Get result . . . . .	2
1.3	Get requests History . . . . .	2
<b>2</b>	<b>User Management</b>	<b>2</b>
2.1	Login . . . . .	2
2.2	Log out . . . . .	3
2.3	Sign Up . . . . .	3
2.4	Update user's Information . . . . .	3
<b>3</b>	<b>File/Specie Upload</b>	<b>3</b>
3.1	Check upload . . . . .	3
3.2	Upload file . . . . .	3
3.3	Import . . . . .	4
3.4	Viewmyfiles . . . . .	4
3.5	Viewmyspecies . . . . .	4
3.6	Deletemyfiles . . . . .	4
3.7	json.command.txt . . . . .	4
<b>4</b>	<b>josn.txt</b>	<b>5</b>
4.1	Input . . . . .	5
4.2	Input Example . . . . .	5
4.3	Output . . . . .	6
4.4	Output example when ok: . . . . .	7
4.5	Output example when failed: . . . . .	8
4.6	Appendix. Equations . . . . .	8

## 1 Main functions

### 1.1 Submit a Crisper-X request

Submit a Crisper-X request.

URL	getMain.php
See also	json.txt[ section 4 ].

### 1.2 Get result

Get the result given the request id.

URL	getResult.php
POST	id
Return	Result in json style, with status=0, 1, 2 means ok, error and not finished respectively.
Example	POST: id=355 Return: { "status":2, "message":"not finished yet" }

### 1.3 Get requests History

Get requests history. Unregistered users will get the global history.

URL	getHistory.php
POST	token
Return	Top ten records, ordered by time, in Json array style
Example	POST: token=005536bf21179e54370c75b6fb136f2a Return: [{ "request_id":"351", "status":"0" }, { "request_id":"355", "status":"2" }]
See also	Login [ section 2.1 ]

## 2 User Management

### 2.1 Login

Login(Sign in) using Username and Password. A token will be returned for future authentication.

URL	login/
POST	name(username), pswd(password)
Return	32 byte token when login succeed, or '-' otherwise.
Example	POST: name=guest&pswd=1234 Return: 005536bf21179e54370c75b6fb136f2a

## 2.2 Log out

Delete user log information and Log out.

URL	login/logout.php
POST	token
Return	Nothing.
See also	Login [ section 2.1 ]

## 2.3 Sign Up

Create a new account with username, password and email.

URL	login/signup.php
POST	name(username, unique), pswd(password, encrypt using md5), email(unique).
Return	"Sign In Succeed" or error information.

## 2.4 Update user's Information

Update your password and/or email address.

URL	login/signup.php
POST	name(username), pswd(old password), newpswd(new password *), newemail(new email *), * Not necessary
Return	'+' : ok, '-' : failed.

# 3 File/Specie Upload

## 3.1 Check upload

Check if this token can upload files

URL	upload/check.php
POST	status=0(Yes)/1(No) and message if status=1 in JSON.
Return	'+' : ok, '-' : failed.
Example	{"status":1, "message":"Authentication failed"}

## 3.2 Upload file

Upload your file.

URL	upload/
POST	token, filename(no longer than 255), note(no longer than 1000), file
Return	A(Upload succeed) or N(Something goes wrong)

### 3.3 Import

Import data from file(s).

URL	upload/import.php
POST	token, command(in json, see section3.7)
Return	New specie Name(start with number), or 'N' for 'Something goes wrong'

### 3.4 Viewmyfiles

View all my files.

URL	upload/viewmyfiles.php
POST	token
Return	All my files in JSON Array Style
Example	[{"fileName":"NC_001133-chromosome1", "note":"Saccharomyces's chromosome"}]

### 3.5 Viewmyspecies

View all my species.

URL	upload/viewmyspecies.php
POST	token
Return	All my species in JSON Array Style
Example	[ {"specie":"Saccharomyces-cerevisiae", "PAMs":[{"PAM":"NGG"}, {"PAM":"NRG"}], "chromosomes":[{"chromosome":"NC_001147-chromosome1"}, ... {"chromosome":"NC_001147-chromosome16"} ]

### 3.6 Deletemyfiles

Delete your file.

URL	upload/deletemyfiles.php
POST	token, filename
Return	A>Delete succeed) or N(Something goes wrong)

### 3.7 json.command.txt

How to write commands in Import(section 3.3) method.

The structure of a command:

```

{
  specie      varchar(96)   Name of the new Specie.
  files       array         All correlative chromosome files(.fna).
  [
    fileName  varchar(255)  File name
  ]
  PAM         varchar(20)   PAM
}

```

Example:

```

{"specie":"testSpecie","files":[{"fileName":"NC_012971"}],"PAM":"NGG"}

```

## 4 josn.txt

### 4.1 Input

type	1	1=Knockout
specie	E.coil	Specie
gene	thrA	Gene Name
location	Chr2:336..2798	Location
pam	NGG	PAM Sequence [1]
r1	0.65	r1, r2=1-r1
length	20	nt length
region	"0000"	Region filter. 1 for in and 0 for out. EXON,INTRON,UTR,INTERGENIC
rfc	"000000"	1 for use and 0 for not. RFC: 10, 12, 12a, 21, 23, 25

### 4.2 Input Example

```

{
  "specie"      : "Saccharomyces-cerevisiae",
  "location"    : "NC_001144-chromosome12:1..500",
  "pam"         : "NGG",
  "rfc"         : "100101"
}

```

### 4.3 Output

status	0	Request Status. (0: ok, 1: failed, 2: still running)
message	no args	Return message
specie	E.coil	Specie
gene	thrA	Gene name
location	Chr2:336..2798	Location
region	array[]	Selected sequence information
{		
endpoint	1807	Region end point
description	intergenic	Region information
}		
result	array[]	Top 50 Only
{		
grna	TC...CGG(20)	sgRNA sequence
position	Chr2:15413205	Location
strand	+	Which DNA chain (+/-)
region	exon	Region of this sgRNA
total_score	86	Sguide
Spe	93	Spe
Seff	7	Seff
count	2	Nmm
offtarget	array[]	Top 20 Only
{		
osequence	CT...TGGG(20)	possible-offtarget sgRNA sequence
oscore	3.5	Smm
omms	4	Nmm
oposition	Chr3:4158	Location of po-sgRNA
ostrand	+	Which DNA chain (+/-)
oregion	Intergenic	Region of this sgRNA
}		
}		

#### 4.4 Output example when ok:

```
{
  "status":      0,
  "message":     {
    "specie":    "E.coli",
    "kind":      "E.coli K12-MG1655",
    "gene":      "",
    "location":  "1:336..2798"
  },
  "result":      [{ "key": "#1",
    "grna":      "GAAGTTCGGCGGTACATCAGTGG",
    "position":  "1:368",
    "total_score": 100,
    "count":     0,
    "offtarget": []
  }, {
    "key":      "#2",
    "grna":      "TAATGAAAAAGGCGAACTGGTGG",
    "position":  "1:935",
    "total_score": 100,
    "count":     0,
    "offtarget": []
  }, {
    "key":      "#3",
    "grna":      "TGGAAAGCAATGCCAGGCAGGGG",
    "position":  "1:427",
    "total_score": 100,
    "count":     0,
    "offtarget": []
  }, {
    "key":      "#4",
    "grna":      "CAAAATCACCAACCACCTGGTGG",
    "position":  "1:479",
    "total_score": 100,
    "count":     0,
    "offtarget": []
  }
],
  .....
}
```

```

{
  "key":          "#44",
  "grna":         "ATTTTGGCCGAACTTTGGACGGG",
  "position":     "1:564",
  "total_score":  91,
  "count":        2,
  "offtarget":    [{
    "osequence":  "ATTTTCGCCAAACATTTGGCAGG",
    "oscore":     0.954750,
    "omms":       4,
    "ostrand":    "_",
    "oposition":  "1:1924344",
    "oregion":    "exco"
  }], {
    "osequence":  "ATTGTTGCGCAACTTTTGGCTGG",
    "oscore":     0.480417,
    "omms":       4,
    "ostrand":    "_",
    "oposition":  "1:3827949",
    "oregion":    "exco"
  }]
}

```

#### 4.5 Output example when failed:

```

{
  "status":      1,
  "message":     "illegal args"
}

```

#### 4.6 Appendix. Equations

R=A,G;    M=A,C;    W=A,T;    S=C,G;    K=G,T;    Y=C,T;  
 H=A,C,T;   V=A,C,G;   B=C,G,T;   D=A,G,T;   N=A, G, C, T